Implementasi Keamanan Transmisi Citra Digital di Jaringan Komputer dengan Kriptografi Blum Blum Shub

Artikel Ilmiah

Peneliti:
Patlas Deo Hani(672010610)
Evangs Mailoa, S.Kom. M.Cs.

Program Studi Teknik Informatika
Fakultas Teknologi Informasi
Universitas Kristen Satya Wacana
Salatiga
Januari 2016

Lembar Pengesahan

Judul Tugas Akhir	: Implementasi Keamanan Transmisi Citra Digital di Jaringan Komputer dengan Kriptografi Blum Blum Shub
Nama Mahasiswa	: Patlas Deo Hani
NIM	: 672010610
Program Studi	: Teknik Informatika
Fakultas	: Teknologi Informasi

Menyetujui,

Evangs Mailoa, S.Kom., M.Cs.
Pembimbing

Mengesahkan,

Dr. Dharmaputra T. Palekahelu, M.Pd.
Dekan

Suprihadi, S.Si., M.Kom.
Ketua Program Studi

Dinyatakan Lulus Ujian tanggal: 4 Februari 2016

Penguji:

1. Prof. Dr. Ir. Eko Sediyono, M.Kom.
2. Alz Danny Wowor, S.Si., M.Cs.

Implementasi Keamanan Transmisi Citra Digital di Jaringan Komputer dengan Kriptografi Blum Blum Shub

Oleh,

Patlas Deo Hani
NIM : 672010610

ARTIKEL ILMIAH

Diajukan Kepada Program Studi Teknik Informatika guna memenuhi sebagian dari persyaratan untuk mencapai gelar Sarjana Komputer

Disetujui oleh,

Evangs Mailoa, S.Kom., M.Cs.
Pembimbing

Diketahui oleh,

Dr. Dharmaputra T. Palekahelu, M.Pd.
Dekan

Suprihadi, S.Si., M.Kom.
Ketua Program Studi

FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS KRISTEN SATYA WACANA
SALATIGA
2016

FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS KRISTEN SATYA WACANA
Jalan Diponegoro 52 – 60
Phone. (0298) 321212 (Hunting)
Fax. (0298) 321433
E-mail: fti@uksw.edu
Salatiga 50711 – INDONESIA

LEMBAR PERSETUJUAN PUBLISH JURNAL

Dengan mempertimbangkan isi dari jurnal mahasiswa :

Nama Mahasiswa : PAILAS DEO HANI
NIM : 672010610

Maka jurnal ini dinyatakan :

~~LAYAK TERBIT~~ / TIDAK LAYAK TERBIT

Menyetujui,

(.....)
Pembimbing 1

(.....)
Pembimbing 2

Mengetahui,

(.....)
Penguji 1

(.....)
Penguji 2

PERPUSTAKAAN UNIVERSITAS
UNIVERSITAS KRISTEN SATYA WACANA
Jl. Diponegoro 52 – 60 Salatiga 50711
Jawa Tengah, Indonesia
Telp. 0298 – 321212, Fax. 0298 321433
Email: library@adm.uksw.edu ; http://library.uksw.edu

PERNYATAAN PERSETUJUAN AKSES

Saya yang bertanda tangan di bawah ini:

Nama : PATLAS DEO HANI
NIM : 67201610 Email : deo.patlas@gmail.com
Fakultas : FTI Program Studi : TEKNIK INFORMATIKA
Judul tugas akhir : IMPLEMENTASI KEAMANAN TRANSMISI CITRA DIGITAL DI JARINGAN KOMPUTER DENGAN KRIPTOGRAFI BLUM BLUM SHUB

Dengan ini saya menyerahkan hak *non-eksklusif** kepada Perpustakaan Universitas – Universitas Kristen Satya Wacana untuk menyimpan, mengatur akses serta melakukan pengelolaan terhadap karya saya ini dengan mengacu pada ketentuan akses tugas akhir elektronik sebagai berikut (beri tanda pada kotak yang sesuai):

☑ a. Saya mengijinkan karya tersebut diunggah ke dalam aplikasi Repositori Perpustakaan Universitas, dan/atau portal GARUDA

☐ b. Saya tidak mengijinkan karya tersebut diunggah ke dalam aplikasi Repositori Perpustakaan Universitas, dan/atau portal GARUDA**

* *Hak yang tidak terbatas hanya bagi satu pihak saja. Pengajar, peneliti, dan mahasiswa yang menyerahkan hak non-ekslusif kepada Repositori Perpustakaan Universitas saat mengumpulkan hasil karya mereka masih memiliki hak copyright atas karya tersebut.*

** *Hanya akan menampilkan halaman judul dan abstrak. Pilihan ini harus dilampiri dengan penjelasan/ alasan tertulis dari pembimbing TA dan diketahui oleh pimpinan fakultas (dekan/kaprodi).*

Demikian pernyataan ini saya buat dengan sebenarnya.

Salatiga, 23 Februari 2016

Patlas Deo Hani
Tanda tangan & nama terang mahasiswa

Mengetahui,

Evongs Mailoa, S.Kom, M.Cs.
Tanda tangan & nama terang pembimbing I

Tanda tangan & nama terang pembimbing II

F-LIB-081

PERPUSTAKAAN UNIVERSITAS
UNIVERSITAS KRISTEN SATYA WACANA
Jl. Diponegoro 52 – 60 Salatiga 50711
Jawa Tengah, Indonesia
Telp. 0298 – 321212, Fax. 0298 321433
Email: library@adm.uksw.edu ; http://library.uksw.edu

PERNYATAAN TIDAK PLAGIAT

Saya yang bertanda tangan di bawah ini:

Nama : PATLAS DEO HANI

NIM : 67200610 Email : deo.patlas@gmail.com

Fakultas : FTI Program Studi : TEKNIK INFORMATIKA

Judul tugas akhir : IMPLEMENTASI KEAMANAN TRANSMISI CITRA DIGITAL DI JARINGAN KOMPUTER DENGAN KRIPTOGRAFI BLUM BLUM SHUB

Pembimbing : 1. EVANGS MAILOA, S.Kom., M.Cs.

2. _____

Dengan ini menyatakan bahwa:

1. Hasil karya yang saya serahkan ini adalah asli dan belum pernah diajukan untuk mendapatkan gelar kesarjanaan baik di Universitas Kristen Satya Wacana maupun di institusi pendidikan lainnya.
2. Hasil karya saya ini bukan saduran/terjemahan melainkan merupakan gagasan, rumusan, dan hasil pelaksanaan penelitian/implementasi saya sendiri, tanpa bantuan pihak lain, kecuali arahan pembimbing akademik dan narasumber penelitian.
3. Hasil karya saya ini merupakan hasil revisi terakhir setelah diujikan yang telah diketahui dan disetujui oleh pembimbing.
4. Dalam karya saya ini tidak terdapat karya atau pendapat yang telah ditulis atau dipublikasikan orang lain, kecuali yang digunakan sebagai acuan dalam naskah dengan menyebutkan nama pengarang dan dicantumkan dalam daftar pustaka.

Pernyataan ini saya buat dengan sesungguhnya. Apabila di kemudian hari terbukti ada penyimpangan dan ketidakbenaran dalam pernyataan ini maka saya bersedia menerima sanksi akademik berupa pencabutan gelar yang telah diperoleh karena karya saya ini, serta sanksi lain yang sesuai dengan ketentuan yang berlaku di Universitas Kristen Satya Wacana.

Salatiga, 23 Februari 2016

Tanda tangan & nama terang mahasiswa

Patlas Deo Hani

F-LIB-080

Implementasi Keamanan Transmisi Citra Digital di Jaringan Komputer dengan Kriptografi Blum Blum Shub

[1] Patlas Deo Hani,[2]Evangs Mailoa
Fakultas Teknologi Informasi
Universitas Kristen Satya Wacana
Jl. Diponegoro 52-60, Salatiga 50711, Indonesia
Email: [1]deo.patlas@gmail.com,[2]evangs.mailoa@staff.uksw.edu

Abstract

Digital image, as well as other information that is passed on the Internet, have an increased risk for information leaks, damage information, and manipulation of information. One technique in securing information is cryptography. Cryptography works by changing the original information, into data that can not be understood. The confidentiality of the information in it depends on the secrecy of the key used for encryption. Blum Blum Shub algorithm is an algorithm CSPRNG that can be used to generate random number row, which is safe for use in the encryption process. In this study developed cryptographic applications for digital imagery, using Blum Blum Shub algorithm for generating keys. The test results showed that, Blum Blum Shub produce cipher image that have different pixel values up to 100%. The resulting application able to send cipher images via LAN, and can be received and decrypted by the receiving party.

Keywords: *Crypthography, CSPRNG, Blum Blum Shub*

Abstrak

Citra digital, seperti halnya informasi lain yang dilewatkan di Internet, memiliki resiko untuk mengalami kebocoran informasi, kerusakan informasi, dan manipulasi informasi. Salah satu teknik dalam mengamankan informasi adalah kriptografi. Kriptografi bekerja dengan cara mengubah informasi semula, menjadi data-data yang tidak dapat dimengerti. Kerahasiaan informasi didalamnya bergantung dari kerahasiaan kunci yang digunakan untuk proses enkripsi. Algoritma Blum Blum Shub merupakan algoritma CSPRNG yang dapat digunakan untuk membangkitkan deretan bilangan acak, yang aman untuk digunakan pada proses enkripsi. Pada penelitian ini dikembangkan aplikasi kriptografi untuk citra digital, dengan menggunakan Algoritma Blum Blum Shub sebagai pembangkit kunci. Hasil pengujian menunjukkan bahwa dengan algoritma Blum Blum Shub, diperoleh *cipherimage* yang memiliki perbedaan nilai piksel sampai dengan 100%. Aplikasi yang dihasilkan dapat mengirimkan cipher image melalui LAN, dan dapat diterima dan didekripsi oleh pihak penerima.

Kata Kunci: Kriptografi, CSPRNG, Blum Blum Shub

[1]Mahasiswa Program Studi Teknik Informatika, Fakultas Teknologi Informasi, Universitas Kristen Satya Wacana
[2,]Staf Pengajar Fakultas Teknologi Informasi, Universitas Kristen Satya Wacana.

1. Pendahuluan

Kemajuan teknologi komunikasi mempermudah proses pertukaran informasi. Informasi dapat dalam bentuk teks, citra digital, video, audio, maupun dalam format data yang lain. Citra digital digunakan untuk mengirimkan informasi seperti hasil survey lapangan, rekam medik, fotografi, dan lain sebagainya. Berdasarkan tiap kegunaan tersebut, citra digital penting untuk dijaga kerahasiaan dan keutuhannya.

Citra digital, seperti halnya informasi lain, memiliki resiko untuk mengalami kebocoran informasi, kerusakan informasi, dan manipulasi informasi. Salah satu teknik dalam mengamankan informasi adalah kriptografi. Kriptografi bekerja dengan cara mengubah informasi semula, menjadi data-data yang tidak dapat dimengerti. Kerahasiaan informasi didalamnya bergantung dari kerahasiaan kunci yang digunakan untuk proses enkripsi.

Algoritma Blum Blum Shub (BBS) merupakan algoritma CSPRNG (*Cryptographically Secure Pseudo Random Number Generator*), yang dapat digunakan untuk membangkitkan deretan bilangan acak, yang aman untuk digunakan pada proses enkripsi. CSPRNG menghasilkan deret bilangan yang lebih acak daripada PRNG. Karena kelebihan tersebut, maka deretan bilangan acak tersebut lebih aman ketika digunakan sebagai kunci enkripsi.

Pada penelitian ini dikembangkan aplikasi kriptografi untuk citra digital, dengan menggunakan Algoritma Blum Blum Shub sebagai pembangkit kunci. Kunci yang dibangkitkan kemudian digunakan pada algoritma kriptografi Vernam [1].Analisis akan dilakukan pada perbandingan hasil enkripsi antara algoritma Blum Blum Shub dengan algoritma PRNG. Diharapkan dengan algoritma Blum Blum Shub akan diperoleh *cipherimage* yang lebih acak, dengan waktu proses yang secara signifikan tidak jauh dari waktu proses PRNG biasa.

2. Tinjauan Pustaka

Pada penelitian Kurnia [2], dirancang aplikasi kriptografi dengan menggunakan algoritma Caesar *cipher* dan PRNG. Pada penelitian tersebut, dibahas tentang masalah keamanan citra digital, yang dikirimkan lewat jaringan komputer dan internet memiliki resiko untuk diketahui, diubah dan dirusak oleh pihak tertentu. Teknik kriptografi dapat diterapkan untuk melindungi citra digital dari masalah-masalah tersebut, salah satunya dengan menggunakan Caesar *cipher*. Caesar *cipher* merupakan algoritma kriptografi klasik yang bekerja dengan cara menggeser karakter sebesar nilai angka tertentu. Untuk meningkatkan keamanan Caesar *cipher*, angka yang digunakan untuk pergeseran dibuat bervariasi. Pada penelitian tersebut dihasilkan aplikasi enkripsi dan dekripsi citra digital dengan menggunakan Caesar *cipher* termodifikasi. Hasil penelitian menunjukkan bahwa pengamanan citra digital dapat dilakukan dengan algoritma Caesar *cipher*, dengan meningkatkan keamanan dengan cara menggunakan pergeseran yang semi acak.

Pada penelitian Sinha [3] Blum Blum Shub dikombinasikan dengan Vigenere *cipher* untuk mengamankan informasi. Pada penelitian tersebut, menjelaskan tentang masalah keamanan informasi. Sinha mengajukan metode

penyelesaian berupa penggabungan teknik kriptografi dengan steganografi. Informasi disandikan dengan algoritma Vigenere. Hasil enkripsi kemudian disisipkan ke dalam *file* audio. Langkah selanjutnya adalah mengacak (*scramble*) *byte* pada *file* audio dengan menggunakan Blum Blum Shub. *Scrambled* stego audio yang merupakan hasil akhir proses, kemudian dikirimkan ke pihak penerima.

Berdasarkan penelitian-penelitian yang telah dilakukan tentang kriptografi dengan PNRG, dan Blum Blum Shub (BBS), maka dilakukan penelitian yang membahas tentang analisis kriptografi citra digital dengan BBS dan LCG. Analisis dilakukan terhadap perbandingan hasil enkripsi antara algoritma BBS sebagai salah satu algoritma dari jenis CSPRNG, dan algoritma LCG sebagai salah satu algoritma dari jenis PRNG. BBS dan LCG digunakan untuk membangkitkan deretan kunci, kemudian kunci tersebut digunakan oleh algoritma Vernam *cipher* untuk melakukan proses enkripsi/dekripsi.

Kriptografi (*cryptography*) merupakan ilmu dan seni penyimpanan pesan, data, atau informasi secara aman. Kriptografi (*Cryptography*) berasal dari bahasa Yunani yaitu dari kata *Crypto* dan *Graphia* yang berarti penulisan rahasia [4]. Kriptografi merupakan bagian dari suatu cabang ilmu matematika yang disebut *Cryptology*. Kriptografi membutuhkan suatu algoritma (*cipher*) dan kunci (*key*) dalam mengenkripsi dan mendekripsi data. *Cipher* adalah fungsi matematika yang digunakan untuk mengenkripsi dan mendekripsi. Sedangkan kunci merupakan sederetan *bit* yang diperlukan untuk mengenkripsi dan mendekripsi data [5]. Secara umum proses kriptografi dibagi menjadi dua bagian yaitu enkripsi dan dekripsi. Data yang telah dienkripsi disebut *ciphertext* karena data asli telah mengalami proses di dalam sebuah algoritma kriptografi atau lebih dikenal dengan nama*cipher*. Proses mengubah pesan yang telah dienkripsi (*ciphertext*) menjadi pesan asli (*plaintext*) disebut sebagai proses dekripsi.

Cryptographically secure pseudorandom number generator (CSPRNG) merupakan salah satu pembangkit bilangan acak yang cocok digunakan untuk kriptografi [6]. CSPRNG memiliki beberapa syarat yaitu lolos uji keacakan statistik dan tahan terhadap serangan yang serius dimana serangan ini bertujuan untuk memprediksi bilangan acak yang dihasilkan.

Ada beberapa jenis dari CSPRNG salah satunya adalah Blum Blum Shub (BBS) [7]. BBS ini dibuat pada tahun 1986 oleh Lenore Blum (distinguished professor of Computer Science at Carnegie Mellon), Manuel Blum (Venezuelan computer scientist) dan Michael Shub (American *mathematician*). Blum Blum Shub ini berbasis teori bilangan. Berikut adalah urutan langkah dari algoritma BBS.

1. Pilih dua bilangan prima p dan q, di mana p dan q keduanya kongruen 3 modulo 4. $p \equiv 3 \bmod 4$ dan $q \equiv 3 (\text{mod })4$.
2. Hasilkan bilangan bulat blum n dengan menghitung $n = p \times q$.
3. Pilih lagi sebuah bilangan acak s sebagai umpan, bilangan yang dipilih harus memenuhi kriteria:
    a. $2 \leq s < n$.
    b. s dan n relatif prima.

4. Hitung nilai $x_0 = s^2 \bmod n$.
5. Hasilkan bilangan bit acak dengan cara :
    a. Hitung $x_i = x_{i-1}^2 \bmod n$.
    b. Hasilkan zi = bit - bit yang diambil dari x_i. Bit yang diambil bisa merupakan LSB (Least Significant Bit) / hanya satu bit atau sebanyak j bit (j tidak melebihi $\log_2 (\log_2 n)$).
6. Bilangan bit acak dapat digunakan langsung atau diformat dengan aturan tertentu, sedemikian hingga menjadi bilangan bulat.

Vernam Cipher diciptakan olehMayor J. Maugborne dan G. Vernam pada tahun 1917. Metode tersebut juga dikenal dengan nama *One Time Pad* (OTP). *Vernam Cipher* merupakan algoritma berjenis kriptografi simetris. Pada proses enkripsi, *plaintext* diubah ke dalam kode ASCII dan kemudian dikenakan operasi XOR terhadap kunci yang sudah diubah ke dalam kode ASCII. Pada proses dekripsi, *ciphertext* diubah ke dalam kode ASCII, kemudian dikenakan operasi XOR terhadap kunci yang sudah diubah ke dalam kode ASCII juga. Contoh enkripsi *Vernam Cipher*, dengan pesan"HELLO", dan kunci enkripsi yang digunakan adalah "XMCKL".

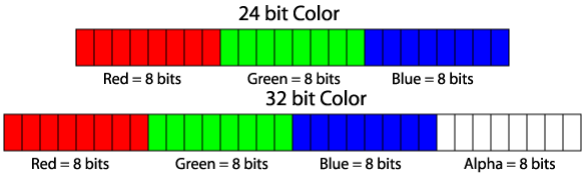
Tabel 1 Contoh Enkripsi *Vernam Cipher*

Pesan	**Pesan ASCII**		**Kunci**	**Kunci ASCII**		**Hasil XOR ASCII**	
	DEC	HEX		DEC	HEX	DEC	HEX
H	72	48	X	88	58	16	10
E	69	45	M	77	4D	8	8
L	76	4C	C	67	43	15	F
L	76	4C	K	75	4B	7	7
O	79	4F	L	76	4C	3	3

Perlu diperhatikan, pada Tabel 1, hasil enkripsi dalam angka ASCII mewakili karakter yang tidak bisa dicetak/ditampilkan (*non-printable character*), sebagai contoh kode ASCII 8 mewakili karakter "*Backspace*" dan kode ASCII 7 mewakili perintah "Bell" [8]. Kedua proses tersebut memiliki langkah yang sama, perbedaan hanya pada *input* dan *output* saja.

Vernam Cipher dipilih karena beberapa alasan. *Vernam Cipher* merupakan *streamcipher*, sehingga memiliki jaminan bahwa panjang *byteplaintext*sama dengan panjang *ciphertext*.*Vernam Cipher* bekerja dengan cara yang sederhana, yaitu dengan menggunakan operator XOR, sehingga waktu proses enkripsi dan dekripsi menjadi cepat.Jika digunakan dengan tepat, yaitu menggunakan kunci yang berbeda-beda atau acak, *Vernam Cipher* tidak akan bisa dipatahkan.

Pada penelitian ini, format *file* citra *digital* yang digunakan adalah PNG. PNG merupakan *TrueColorimage*, yang berarti tiap piksel direpresentasikan dengan 3 *byte*, terbagi ke dalam *red*, *green*, dan *blue* masing-masing 1 *byte*. Hal ini sering disebut dengan warna RGB, atau *TrueColor24bit*. Selain 24 *bit* warna, *file*PNG juga mendukung 32 *bit* warna. *TrueColor* 32 *bit* sama dengan 24 *bit*, dengan perbedaan adalah adanya 1 *byte* tambahan yang disebut komponen *alpha*[9].



Gambar1*TrueColor24bit* dan *32bit*

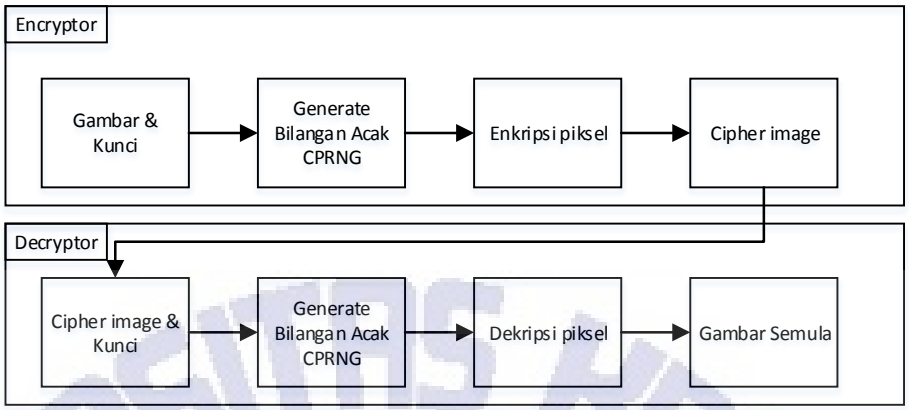
Gambar 1 menunjukkan gambaran susunan *bit* pada true color *image*[10]. Pada 24 *bit* warna terdapat 8 *bit* komponen warna merah, 8 *bit* warna hijau, dan 8 *bit* warna biru. Pada 32 *bit* warna, terdapat tambahan 8 *bit* untuk komponen *alpha*, yang berfungsi untuk transparansi gambar.

3. Metode dan Perancangan Sistem

Penelitian yang dilakukan, diselesaikan melalui tahapan penelitian yang terbagi dalam empat tahapan, yaitu: (1) Identifikasi masalah dan studi literatur, (2) Perancangan sistem, (3) Perancangan *Prototype* sistem, dan (4) Pengujian *prototype* sistem dan analisis hasil pengujian.

Gambar2 Tahapan Penelitian

Tahapan penelitian pada Gambar 2, dapat dijelaskan sebagai berikut. *Tahap pertama*: identifikasi masalah, yaitu rentannya informasi yang dilewatkan di Internet, dan diperlukan mekanisme untuk mengamankan informasi tersebut; *Tahap kedua:* perancangan sistem yang meliputi perancangan proses enkripsi dan proses dekripsi; *Tahap ketiga*: perancangan *prototype* sistem, yaitu membuat aplikasi sesuai perancangan proses pada tahap kedua; dan *Tahap keempat:* pengujian sistem dan analisis hasil pengujian, yaitu dilakukan pengujian terhadap proses yang telah dirancang, dan melihat kesesuaian solusi terhadap masalah yang telah teridentifikasi sebelumnya.



Gambar 3 Rancangan Aristektur Sistem

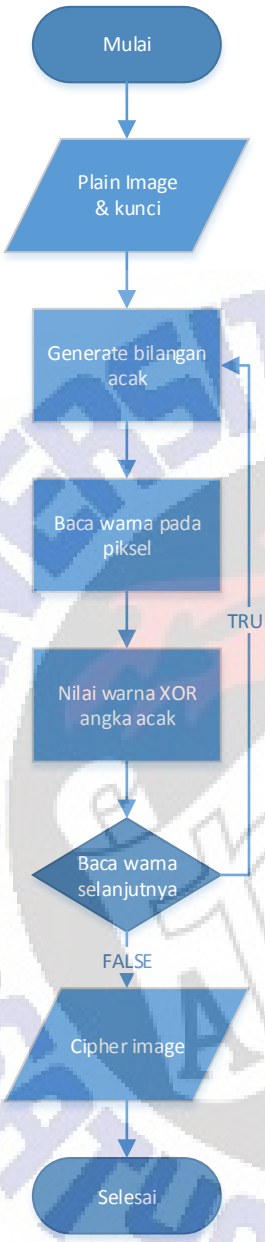
Sistem dikembangkan dalam bentuk aplikasi desktop. Ada dua sisi pengguna, yaitu sisi *encryptor*, dan sisi *decryptor*. *Encryptor* bertindak sebagai pemilik informasi, dan bertujuan untuk mengirim informasi dalam bentuk gambar, kepada *decryptor*. *Encryptor* menggunakan satu kunci pada proses enkripsi, dan kunci yang sama juga harus diketahui oleh *decryptor* untuk digunakan pada proses dekripsi.

Proses enkripsi (Gambar 4) terdiri dari beberapa subproses. Proses pertama adalah pembangkitan bilangan acak dengan menggunakan kunci yang dimasukkan oleh pengguna. Proses kedua adalah proses pembacaan piksel gambar. Proses ketiga adalah enkripsi piksel dengan menggunakan bilangan acak sebagai kunci.

Proses pembangkitan bilangan acak dilakukan dengan menggunakan algoritma Blum Blum Shub. Untuk mendapatkan urutan yang dapat ditelusuri kembali ketika proses dekripsi, diperlukan satu nilai sebagai *seed*. Pada penelitian ini nilai *seed* diperoleh dengan cara menjumlahkan nilai *byte* dari tiap-tiap karakter pada kunci yang dimasukkan oleh pengguna. Bilangan acak yang dibangkitkan, berjumlah sesuai dengan kebutuhan panjang kunci enkripsi.

Setelah bilangan acak diperoleh, maka dilakukan proses membaca piksel. Pada piksel diperoleh ketiga nilai warna RGB (*red, green, blue*). Nilai warna tersebut kemudian dilakukan operasi XOR dengan nilai bilangan acak.

Seperti halnya pada proses enkripsi, proses dekripsi (Gambar 5) terdiri dari tiga subproses. Proses pertama adalah pembangkitan bilangan acak dengan menggunakan kunci yang dimasukkan oleh pengguna. Proses kedua adalah proses pembacaan piksel gambar. Proses ketiga adalah dekripsi piksel dengan menggunakan bilangan acak sebagai kunci. Proses pembangkitan bilangan acak dilakukan dengan menggunakan algoritma Blum Blum Shub. Setelah bilangan acak diperoleh, maka dilakukan proses membaca piksel. Pada piksel diperoleh ketiga nilai warna RGB (*red, green, blue*). Nilai warna tersebut kemudian dilakukan operasi XOR dengan nilai bilangan acak.



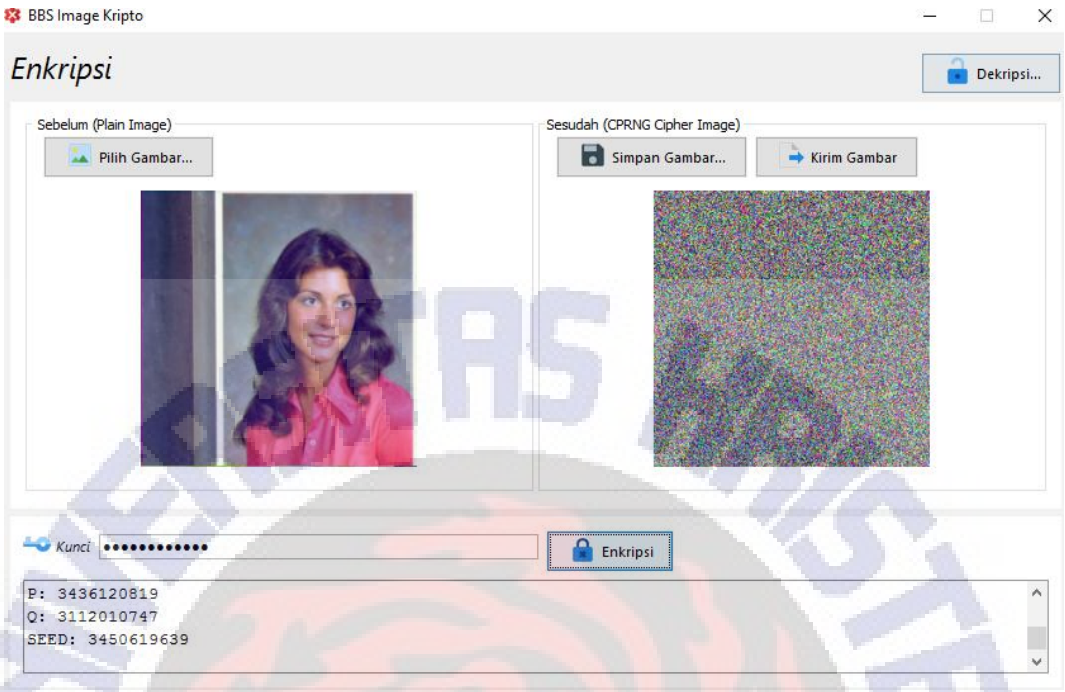
Gambar 4 Rancangan Proses Enkripsi

Gambar 5 Rancangan Proses Dekripsi

Pengujian yang dilakukan adalah dengan membandingkan kecepatan proses dan keamanan informasi dan keberhasilan dalam pengiriman cipher image pada LAN.

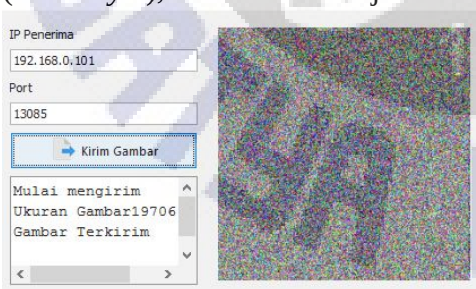
4. Hasil dan Pembahasan

Pada bagian ini dijelaskan tentang hasil penelitian yang telah dilakukan. Sistem diimplementasikan dalam bentuk aplikasi desktop. Pengembangan aplikasi dilakukan dengan menggunakan NetBeans IDE. Aplikasi terdiri dari dua bagian utama, yaitu bagian enkripsi dan bagian dekripsi.

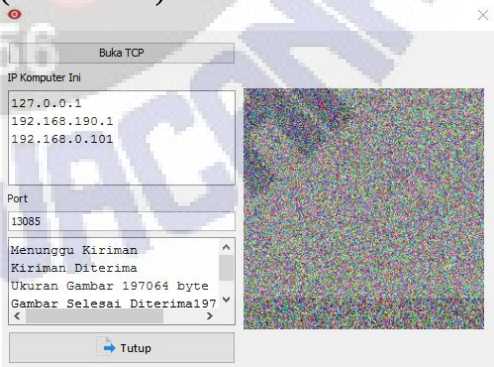


Gambar 6*Form*Enkripsi

Pada Gambar 6 ditunjukkan *form*dekripsi. *Form* tersebut menyediakan kontrol untuk memilih gambar *PNG*, dan *input* kunci. Hasil enkripsiditampilkan setelah proses selesai. Kunci diberi batasan yaitu minimal panjang 12 karakter, maksimal 16. Batas minimal 12 dipilih karena berdasarkan penelitian Marinakis [11], dijelaskan bahwa aturan praktis panjang kunci, untuk melindungi dari serangan *brute force attack* adalah dengan menambahkan 1 *bit* tiap tahun. Marinakis menyebutkan bahwa pada tahun 2012 panjang kunci yang aman adalah 90 *bit*, sehingga untuk saat ini (2016), panjang kunci yang aman adalah 94 *bit* (11.75 *byte*), dibulatkan menjadi 12 *byte* (12 karakter).

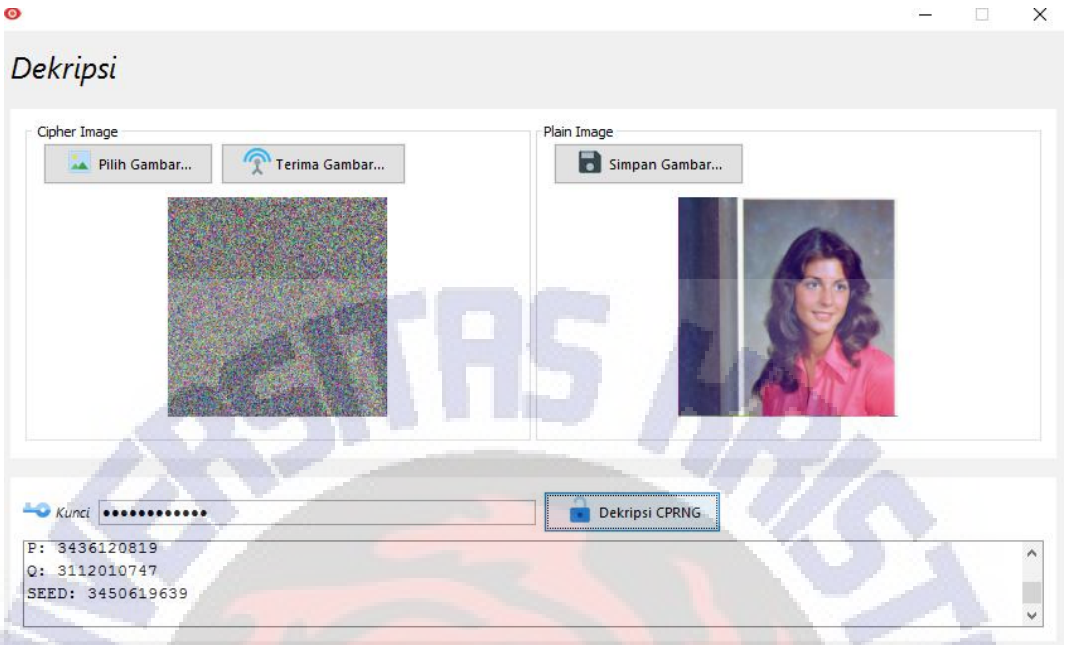


Gambar 7 Form Kirim Gambar



Gambar 8 Form Terima Gambar

Gambar yang telah dienkripsi, dapat disimpan, atau dikirimkan ke penerima dengan menggunakan form yang ditunjukkan pada Gambar 7. Oleh penerima, gambar diterima dengan menggunakan form seperti pada Gambar 8. Penerima kemudian dapat melakukan proses dekripsi dengan menggunakan form dekripsi (Gambar 9).



Gambar 9*Form*Dekripsi

Pada Gambar 9, ditunjukkan *form* yang digunakan untuk proses dekripsi. *Form* ini memerlukan *input* berupa gambar dan kunci.*Output* dari proses dekripsi adalah gambar plain, yaitu gambar hasil proses dekripsi (gambar semula).

Kode Program 1 Perintah untuk mencatat waktu proses

```
1.  long begin = System.currentTimeMillis();
2.  long end = System.currentTimeMillis();
3.  long span = endPRNG - beginPRNG;
4.  this.jTextArea1.append("Waktu Proses " + span + " ms");
```

Waktu proses pada Gambar 7, diperoleh dengan cara menghitung selisih antara waktu mulai dengan waktu selesai. Kode Program 1 menunjukkan cara untuk mencatat waktu proses. Waktu "saat ini" diperoleh dengan menggunakan fungsi "*currentTimeMillis*" pada class "*System*".

Kode Program 2 Perintah untuk membangkitkan kunci dengan CSPRNG Blum Blum Shub

```
5.  public static byte[] generateKey(String feed, int length) {
6.          byte[] feedBytes = feed.getBytes();
7.          long total = 0;
8.          for(byte b: feedBytes){
9.                  total+=b;
10.         }
11.         Random rand= new Random(total);
12.         double p = BBSLibrary.getRandomNumber(rand).doubleValue();
13.         double q = BBSLibrary.getRandomNumber(rand).doubleValue();
14.         double seed = BBSLibrary.getRandomNumber(rand).doubleValue();
15.         BBSTool b = new BBSTool(p, q, seed);
16.
17.         byte[] kunci = new byte[length];
18.         for (int i = 0; i < kunci.length; i++) {
19.                 double x = b.getrandom();
20.                 byte by = (byte)(255 * x);
21.                 kunci[i] = by;
22.         }
23.         return kunci;
24. }
```

Kode Program 2 merupakan perintah untuk proses membangkitkan kunci. Jumlah kunci yang dibangkitkan sama dengan jumlah *byte* warna yang akan

dienkripsi. Sehingga dengan demikian, tiap warna akan dienkripsi dengan 1 *byte* kunci yang berbeda. Algoritma Blum Blum Shub meminimalkan kemungkinan muncul angka acak yang kembar. Kemungkinan tetap masih ada, tapi jauh lebih kecil daripada algoritma jenis PRNG.

Kode Program 3 Perintah untuk proses enkripsi dengan algoritma Blum Blum Shub

```
1.  public static byte[] encrypt(byte[] data, byte[] key) {
2.          byte[] result  = new byte[data.length];
3.          for(int i=0;i<result.length;i++){
4.                  result[i] = (byte)(data[i] ^ key[i%key.length]);
5.          }
6.          return result;
7.  }
```

Kode Program3 merupakan perintah untuk proses enkripsi. Enkripsi dilakukan dengan melakukan operasi XOR antara elemen pada *array* data dengan elemen pada *arraykey*. Panjang data dan panjang *key* sama.

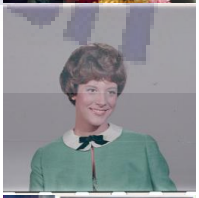
Kode Program 4 Perintah untuk proses dekripsi dengan algoritma Blum Blum Shub

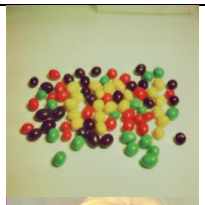
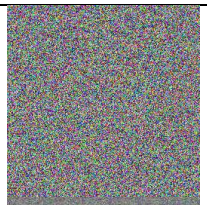
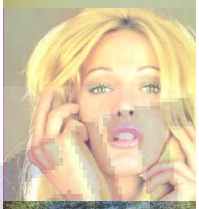
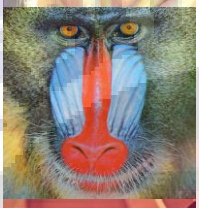
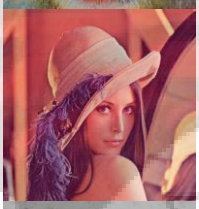
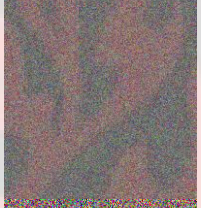
```
1.  public static byte[] decrypt(byte[] data, byte[] key) {
2.          byte[] result  = new byte[data.length];
3.          for(int i=0;i<result.length;i++){
4.                  result[i] = (byte)(data[i] ^ key[i%key.length]);
5.          }
6.          return result;
7.  }
```

Kode Program 4 merupakan perintah proses dekripsi. Baris perintah pada proses dekripsi sama dengan proses enkripsi, yaitu dengan menggunakan operator XOR.

Pengujian pertama yaitu perbedaan piksel antara gambar semula dengan gambar hasil enkripsi. Hasil pengujian ditunjukkan pada tabel 2. Kunci enkripsi adalah “satyawacana”. Pengujian dilakukan dengan menggunakan *standart test image*s[12] yang biasa digunakan pada pengujian algoritma pengolahan gambar, ditambah dengan dua gambar logo UKSW.

Tabel2 Perbedaan Piksel Setelah Proses Enkripsi

No	Gambar Semula	Jumlah Piksel	*Cipher* Blum Blum Shub	Perbedaan
1.		65536		65536 (100% berbeda)
2.		65536		65536 (100%)
3.		65536		65536 (100%)

4.		65536		65536 (100%)
5.		262144		262144 (100%)
6.		262144		262144 (100%)
7.		262144		262144 (100%)
8.		262144		262144 (100%)
9.		40000		39996 (99.99%)
10.		40000		39996 (99.99%)

Berdasarkan hasil pengujian pertama, diketahui bahwa cipher image yang dihasilkan, memiliki perbedaan piksel sampai dengan 100%.

Pengujian kedua adalah pengujian waktu enkripsi dan dekripsi antara kedua algoritma. Hasil perbandingan ditunjukkan pada Tabel 3. Kunci yang digunakan adalah “satyawacana”.

Tabel 3 Waktu Proses Enkripsi dan Dekripsi

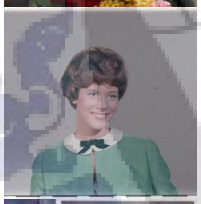
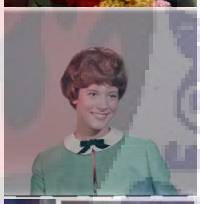
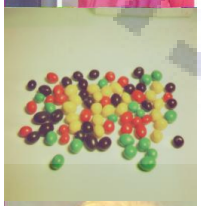
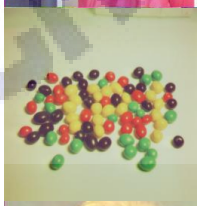
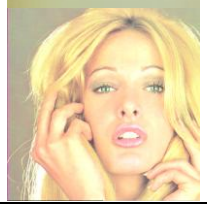
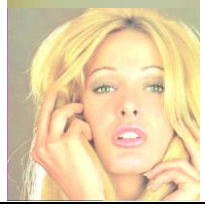
No	Jumlah Piksel	Enkripsi (milidetik)	Dekripsi (milidetik)
1.	65536	28	27
2.	65536	20	20

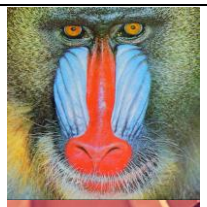
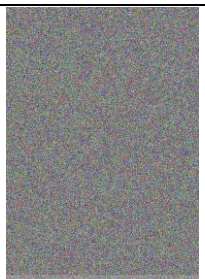
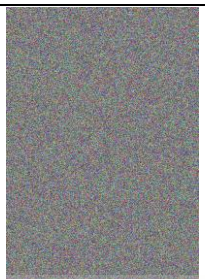
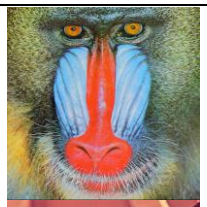
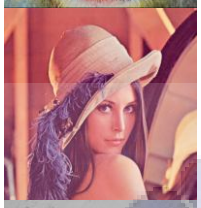
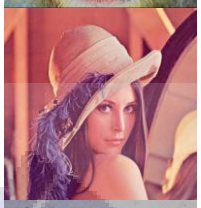
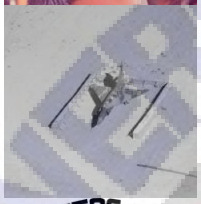
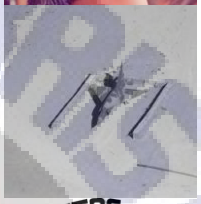
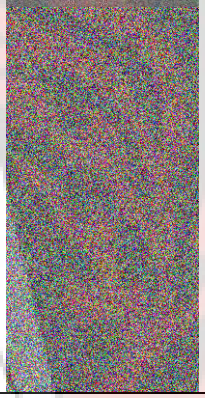
3.	65536	20	20
4.	65536	19	19
5.	262144	70	71
6.	262144	73	73
7.	262144	74	70
8.	262144	73	73
9.	40000	46	48
10.	40000	45	44

Waktu proses enkripsi dan dekripsi tidak terpaut jauh. Pada kedua proses terjadi langkah pembangkitan kunci, dan operasi XOR.

Pengujian ketiga adalah pengujian transfer data melalui jaringan LAN. Pengujian dilakukan dengan cara mengirimkan hasil enkripsi dari pengirim, ke komputer penerima. Penerima akan melakukan dekripsi setelah gambar diterima. Pengujian berhasil jika gambar yang dikirim (gambar asli sebelum enkripsi), sama dengan gambar yang diterima setelah proses dekripsi.

Tabel4 Perbedaan Piksel Setelah Proses Enkripsi

No	Gambar Semula	Gambar yang dikirim	Gambar yang diterima	Gambar Hasil Dekripsi	Kesimpulan
1					Berhasil.
2					Berhasil
3					Berhasil
4					Berhasil
5					Berhasil

6					Berhasil
7					Berhasil
8					Berhasil
9					Berhasil
10					Berhasil

Hasil pengujian ketiga, ditunjukkan pada Tabel 4, memberikan kesimpulan yaitu gambar yang dikirimkan oleh pengirim berada dalam bentuk cipher image. Gambar yang diterima oleh penerima berada dalam bentuk cipher image juga. Kunci yang sama digunakan untuk enkripsi dan dekripsi, sehingga pihak penerima berhasil mendapatkan gambar yang asli.

5. Kesimpulan

Berdasarkan penelitian, pengujian dan analisis terhadap sistem, maka dapat diambil kesimpulan sebagai berikut: (1) Algoritma Blum Blum Shub terbukti aman untuk digunakan dalam menyandikan citra digital, sekalipun proses enkripsi/dekripsi hanya dengan menggunakan operator XOR. Kunci yang acak yang dihasilkan oleh BBS membantu menyandikan nilai-nilai piksel pada gambar, sehingga diperoleh *cipher image* yang memiliki perbedaan nilai piksel sampai dengan 100% dengan nilai piksel gambar semula; (2 Tidak terdapat perbedaan waktu yang signifikan antara proses enkripsi dan dekripsi. Kedua proses tersebut sama-sama terdiri dari proses pembangkitan kunci dengan BBS, dan proses XOR dengan algoritma Vernam; (3) Aplikasi dapat mengirimkan *cipher image* ke penerima melalui LAN. Cipher image yang diterima dapat didekripsi menjadi

gambar semula, dengan menggunakan kunci yang sama untuk proses enkripsi sebelumnya.

Saran yang dapat diberikan untuk penelitian lebih lanjut adalah, perlunya pengujian pada penggunaan memori. Hal ini menjadi sangat penting ketika aplikasi diimplementasikan pada *platform* yang memiliki memori kecil seperti *smartphone*.

6. Daftar Pustaka

[1]. Rosenberg, B. 2004. *Vernam Cipher, A Perfect Cipher*. http://www.cs.miami.edu/home/burt/learning/Csc609.051/notes/02.html. Diakses pada 4 Mei 2015

[2]. Kurnia, D. S. & Pakereng, M. A. I. 2015. *Perancangan dan Implementasi Image Kriptografi Menggunakan Caesar Cipher Termodifikasi*. Program Studi Teknik Informatika Fakultas Teknologi Informasi Universitas Kristen Satya Wacana Salatiga

[3]. Sinha, N., Bhowmick, A. & Kishore, B. 2015. *Encrypted Information Hiding using Audio Steganography and Audio Cryptography*. International Journal of Computer Applications 112.

[4]. Forouzan, B. A. 2007. *Cryptography & Network Security*. McGraw-Hill, Inc.

[5]. Munir, R. 2006. *Kriptografi*. Informatika, Bandung

[6]. Kelly, J. R. 2001. *Cryptographically secure pseudo random number generator*.

[7]. Klein, A. 2013. *The Blum-Blum-Shub Generator and Related Ciphers*. Stream Ciphers

[8]. Juniper Network 2011. *Reference: Nonprintable and Printable ASCII Characters*. http://www.juniper.net/documentation/en_US/idp5.1/topics/reference/general/intrusion-detection-prevention-custom-attack-object-extended-ascii.html. Diakses pada 3 Desember 2015.

[9]. Willamette.edu 2004. *Image File Formats*. www.willamette.edu/~gorr/classes/GeneralGraphics/imageFormats/. Diakses pada 4 Desember 2015.

[10]. Randers-Pehrson, G., Boutell, T. & others 1999. *PNG (Portable Network Graphics) Specification, Version 1.2*. PNG Development Group 28.

[11]. Marinakis, G. 2013. *Minimum key length for cryptographic security Single Search*. Journal of Applied Mathematic & Bioinformatics 3, 181–191.

[12]. Weber, A. 1993. *The USC-SIPI Image Database*. http://sipi.usc.edu/database/. Diakses pada 3 Oktober 2015.